AF
TD

Application No. 10/035,580
Attorney Docket No. 06502.0381-00

PATENT
Customer No. 22,852

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**
**BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

| | |
|---|---|
| In re Application of: | |
| Guy L. STEELE, Jr. | Group Art Unit: 2193 |
| Application No.: 10/035,580 | Examiner: Do, Chat C. |
| Filed: December 28, 2001 | |
| For: FLOATING POINT MULTIPLIER WITH EMBEDDED STATUS INFORMATION | Confirmation No.: 2889 |

**Attention: Mail Stop Appeal Brief-Patents**
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

## APPEAL BRIEF UNDER BOARD RULE § 41.37

In support of the Notice of Appeal filed September 30, 2005, further to Board Rule § 41.37, and in response to the Notification of Non-Compliant Appeal Brief mailed on February 23, 2007, Appellants present this brief. Appellants previously paid the fee of $500.00 required under 37 C.F.R. § 1.17(c), upon originally filing their Appeal Brief on April 7, 2006.

This Appeal responds to the Notice of Panel Decision from Pre-Appeal Brief Review mailed on November 7, 2005 and the Final Office Action mailed on April 8, 2005, which rejected claims 1-40 under 35 U.S.C. § 102(b).

If an additional fee is due for filing this Appeal Brief with the Response to Notification of Non-Compliant Appeal Brief, Appellants request that any such fee be charged to Deposit Account No. 06-0916.

## TABLE OF CONTENTS

## I. REAL PARTY IN INTEREST

Sun Microsystems, Inc. is the real party in interest, as indicated by the assignment in its name, recorded at Reel 012440, Frame 0785 on December 28, 2001.

## II. RELATED APPEALS AND INTERFERENCES

In accordance with 37 C.F.R. § 41.37(c)(1)(ii), Appellant advises the Board of Patent Appeals and Interferences (the "Board") of the following pending appeals, which may be related to, directly affect or be directly affected by, or have a bearing on the Board's decision in the instant appeal:

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,747, in which an Appeal Brief was filed concurrently herewith.

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,595, in which an Appeal Brief was filed concurrently herewith.

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,584, in which an Appeal Brief was filed concurrently herewith.

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,587, in which an Appeal Brief was filed concurrently herewith.

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,647, in which an Appeal Brief was filed concurrently herewith.

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,579, in which an Appeal Brief was filed concurrently herewith.

## III. STATUS OF CLAIMS

Claims 1-40 remain pending and under current examination.

Claims 1-40 have been finally rejected by the Examiner under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,995,991 to Huang et al. (*"Huang"*). Appellant appeals the rejection of those claims. The attached Appendix contains a clean copy of the claims involved in the appeal, claims 1-40.

Claims 1-6, 8-19, 21-32, and 34-40 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-37 of copending Application No. 10/035,584 in view of U.S. Patent No. 5,065,352 to Nakano.

## IV. STATUS OF AMENDMENTS

All amendments have been entered. No amendments under 37 C.F.R. § 1.116 have been filed.

## V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claims 1, 14, and 28 of this application recite a system, method, and computer-readable medium for providing a floating point product. *Specification*, p. 1, paragraph 002.

Digital electronic devices, such as digital computers, calculators and other devices, perform arithmetic calculations on values in integer, or "fixed point," format, in fractional, or "floating point" format, or both. *Specification*, p. 1, paragraph 003. Institute of Electrical and Electronic Engineers (IEEE) Standard 754, (hereinafter "IEEE Std. 754") published in 1985 and adopted by the American National Standards Institute

(ANSI), defines several standard formats for expressing values in floating point format and a number of aspects regarding behavior of computation in connection therewith. *Id.*

In prior art devices that perform floating point computations, floating point status information generated by the computation is stored in a floating point status register. *Id.* at p. 8, paragraph 020. The status information is stored as conditions, represented by flags that are stored in the floating point status register. *Id.* at p. 8, paragraph 026.

However, the modes (*e.g.*, the rounding modes and traps enabled/disabled mode), flags (*e.g.*, flags representing the status information), and traps that are required to implement IEEE Std. 754 introduce implicit serialization issues. *Id.* at p. 9, paragraph 028. Implicit serialization is essentially the need for serial control of access (read/write) to and from globally used registers, such as a floating point status register. *Id.* The potential for implicit serialization makes the Standard difficult to implement coherently and efficiently in today's superscalar and parallel processing architectures without loss of performance. *Id.* at pp. 9-10, paragraph 028.

Moreover, the implicit side effects of a procedure that can change the flags or modes can make it very difficult for compilers to perform optimizations on floating point code. *Id.* at p. 10, paragraph 029. As a result, compilers for most languages usually assume that every procedure call is an optimization barrier in order to be safe. *Id.* This unfortunately may lead to further loss of performance. *Id.*

The claimed invention addresses these and other problems of prior art floating point computational systems. *Id.* at p. 10, paragraph 032. Since the floating point status information comprises part of the floating point representation of the result, instead of being separate and apart from the result as in prior art multiplier units, the

implicit serialization that is required by maintaining the floating point status information separate and apart from the result may be obviated. *Id.* at p. 14, paragraph 045.

The invention, as recited by independent claim 1, relates to a system for providing a floating point product (*Id.* at p. 11, paragraph 034). The system may include an analyzer circuit (Fig. 1, 11A, 11B, 12A, 12B; *Id.* at. P. 19, paragraph 059) configured to determine a first status of a first floating point operand (Fig. 1, 11A) and a second status of a second floating point operand (Fig. 1, 11B) based upon data within the first floating point operand and the second floating point operand respectively (*Id.*; Fig. 2). Further, the system may include a results circuit (Fig. 1, 13, 14, and 15; *Id.*) coupled to the analyzer circuit and configured to assert a resulting floating point operand containing the product of the first floating point operand and the second floating point operand and a resulting status embedded within the resulting floating point operand (*Id.*; *Id.* at p. 14, paragraph 046 (Fig. 2)).

The invention, as recited by independent claim 15, also relates to a method for providing a floating point product (*Id.* at p. 11, paragraph 035). The method may include determining a first status of a first floating point operand (Fig. 1, 11A) and a second status of a second floating point operand (Fig. 1, 11B) based upon within the first floating point operand and the second floating point operand respectively (*Id.*; Fig. 2). Further, the method may include asserting a resulting floating point operand containing the product of the first floating point operand and the second floating point operand and a resulting status embedded within the resulting floating point operand (*Id.*; *Id.* at p. 14, paragraph 046 (Fig. 2)).

The invention, as recited by independent claim 28, further relates to a computer-readable medium on which is stored a set of instructions for providing a floating point product, which when executed perform stages (*Id.* at pp. 10-11, paragraph 036). The executed stages may include determining a first status of a first floating point operand and a second status of a second floating point operand based upon data within the first floating point operand and data within the second floating point operand respectively. *Id.* The executed stages may also include asserting a resulting floating point operand containing the product of the first floating point operand and the second floating point operand and a resulting status embedded within the resulting floating point operand. *Id.*

## VI. GROUNDS OF REJECTION TO BE REVIEWED

A. Claims 1-40 stand rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,995,997 to Huang et al. ("*Huang*"). *See Final Office Action mailed April 8, 2005.* Appellant requests review of this rejection.[1]

## VII. ARGUMENT

### A. Introduction

In view of the reasoning set forth below, Appellant respectfully requests the Board to reverse the Examiner's rejection.

In order to properly establish that *Huang* anticipates Appellant's claimed invention under 35 U.S.C. § 102, each and every element of each of the claims in issue must be found, either expressly described or under principles of inherency, in that single reference. Furthermore, "[t]he identical invention must be shown in as complete detail

1 Pending the withdrawal of the Examiner's rejection under 35 U.S.C. § 102(b), Appellant will obviate the provisional double patenting rejection as appropriate. Appellant does not request the Board to review the double patenting rejection.

as is contained in the ... claim." *See* M.P.E.P. § 2131, quoting *Richardson v. Suzuki Motor Co.*, 868 F.2d 1126, 1236, 9 U.S.P.Q.2d 1913, 1920 (Fed. Cir. 1989).

### B. The rejection of claims 1-40 under 35 U.S.C. § 102(b) as being anticipated by *Huang*

*Huang* does not disclose each and every element of Appellant's claimed invention. Independent claim 1 calls for a combination including, for example,

> an analyzer circuit configured to determine a first status of a first floating point operand and a second status of a second floating point operand based upon data within the first floating point operand and data within the second floating point operand respectively; and
> a results circuit coupled to the analyzer circuit and configured to ... a resulting status embedded within the resulting floating point operand

(emphasis added). *Huang* fails to teach or suggest these elements.

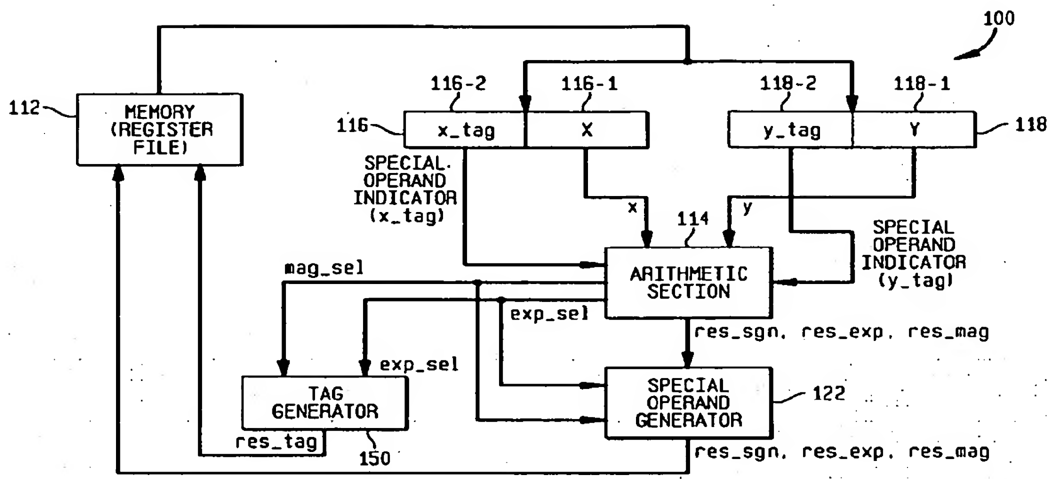
The Examiner asserts that *Huang's* elements 116-2 and 118-2 in Figure 4 constitute the claimed analyzer circuit.[2] *Office Action mailed April 8, 2005* at p. 2. The Examiner further asserts that *Huang's* "x_tag 116 and y_tag 118 [alleged status] are determined based on x and y operands 116-1 and 118-1." *Id.* at p. 9. However, *Huang's* "tag value" (alleged status), is not "data within the first floating point operand and data within the second floating point operand," as recited by claim 1 (emphasis added).

As illustrated in Fig. 4 of *Huang*, which Appellant reproduces below, *Huang* teaches "each portion of the registers 116 and 118 has an operand value storage portion 116-1 and 118-1 and a tag value storage portion 116-2 and 118-2" (emphasis added, *Huang*, col. 6, line 66 through col. 7, line 2).

2 The Examiner also references *Huang's* elements 24 and 26 in Fig. 1. However, Fig. 1 of *Huang* is directed to prior art. If the Examiner continues to rely on the combination of Fig. 1 and Fig. 4 of *Huang*, the rejection should be under 35 U.S.C. § 103(a) and the Examiner must overcome the presumption that Fig. 4 of *Huang* teaches away from the prior art discussed in Fig. 1. *See Huang*, col. 5, lines 3-7.

U.S. Patent Nov. 30, 1999 Sheet 3 of 3 5,995,991

FIG. 4




*Huang* thus teaches a separate operand value storage portion, 116-1, and a separate tag value storage portion for the x_tag 116-2 (*Huang*, col. 6, line 66 through col. 7, line 2; Fig. 4). *Huang's* tag value stored in 116-2 is clearly not "data within" *Huang's* operand value stored in 116-1. Therefore, *Huang* does not teach or suggest at least a system to "determine a first status of a first floating point operand and a second status of a second floating point operand based upon data within the first floating point operand and data within the second floating point operand," as recited by claim 1 (emphasis added).

Moreover, independent claim 1 also recites a combination including, for example, "a resulting status embedded within the resulting … operand," (emphasis added). *Huang* does not teach or suggest at least this additional element.

The Examiner asserts that the output of *Huang's* tag generator 150 (FIG. 4) constitutes the claimed "resulting status embedded within the resulting floating point operand." *Office Action mailed April 8, 2005* at p. 3. However, as illustrated in FIG. 4, the output of *Huang's* tag generator 150 (alleged status) is separate from the output of the special operand generator 122. These results are stored and loaded into the separate tag value and operand values 116-2 and 116-1, as discussed above. Such teachings by *Huang* do not constitute a teaching or suggestion of "a resulting status embedded within the resulting floating point operand," as recited by claim 1 (emphasis added).

Structures such as those taught by *Huang* were acknowledged in the Background section of Appellant's specification, which states "conditions are typically represented by flags that are stored in the floating point status register" (paragraph 026). Claim 1 specifically distinguishes over such structures, calling for determining a first and second status of a first and second operand "based upon data within the first floating point operand and data within the second floating point operand" (emphasis added). Claim 1 also distinguishes over such structures by calling for "a resulting status embedded within the resulting floating point operand" (emphasis added). *See also* Appellant's Figs. 2, 4a, 4b, and 4c.

Because *Huang* does not teach or suggest each and every element recited by claim 1, *Huang* cannot anticipate this claim. Independent claims 15 and 28, although of different scope, recite similar elements to claim 1. Claims 2-14, 16-27, and 29-40 depend from independent claims 1, 15, and 28, and therefore include all of the elements recited therein. Accordingly, for at least the reasons discussed above with respect to

claim 1, *Huang* cannot anticipate claims 2-40. Appellant requests that the Board allow claims 1-40.

## VIII. CONCLUSION

For the reasons given above, the Examiner has not established anticipation with respect to the appealed claims. Accordingly, pending claims 1-40 are allowable and reversal of the Examiner's rejections are respectfully requested.

To the extent any extension of time under 37 C.F.R. § 1.136 is required to obtain entry of this Appeal Brief, such extension is hereby respectfully requested. If there are any fees due under 37 C.F.R. §§ 1.16 or 1.17 which are not enclosed herewith, including any fees required for an extension of time under 37 C.F.R. § 1.136, please charge such fees to our Deposit Account No. 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: March 22, 2007

By: Robert Elam #27432
for Nathan A. Sloan
Reg. No. 56,249

## IX. Claims Appendix to Appeal Brief Under Rule 41.37(c)(1)(viii)

1. A system for providing a floating point product, comprising:

an analyzer circuit configured to determine a first status of a first floating point operand and a second status of a second floating point operand based upon data within the first floating point operand and data within the second floating point operand respectively; and

a results circuit coupled to the analyzer circuit and configured to assert a resulting floating point operand containing the product of the first floating point operand and the second floating point operand and a resulting status embedded within the resulting floating point operand.

2. The system for providing a floating point product of claim 1, wherein the analyzer circuit further comprises:

a first operand buffer configured to store the first floating point operand;

a second operand buffer configured to store the second floating point operand;

a first operand analysis circuit coupled to the first operand buffer, the first operand analysis circuit configured to generate a first characteristic signal having information relating to the first status; and

a second operand analysis circuit coupled to the second operand buffer, the second operand analysis circuit configured to generate a second characteristic signal having information relating to the second status.

3. The system for providing a floating point product of claim 2, wherein the first status and the second status are determined without regard to memory storage external to the first operand buffer and the second operand buffer.

4. The system for providing a floating point product of claim 3, wherein the memory storage external to the first operand buffer and the second operand buffer is a floating point status register.

5. The system for providing a floating point product of claim 1, wherein the results circuit further comprises:

a multiplier circuit coupled to the analyzer circuit, the multiplier circuit configured to produce the product of the first floating point operand and the second floating point operand;

a multiplier logic circuit coupled to the analyzer circuit and configured to produce the resulting status based upon the first status and the second status; and

a result assembler coupled to the multiplier circuit and the multiplier logic circuit, the result assembler configured to assert the resulting floating point operand and embed the resulting status within the resulting floating point operand.

6. The system for providing a floating point product of claim 5, wherein the multiplier logic circuit is organized according to a structure of a decision table.

7. The system for providing a floating point product of claim 1, wherein the product of the first floating point operand and the second floating point operand is identical in all cases to the product that would be produced if the two operands were swapped.

8. The system for providing a floating point product of claim 1, wherein the first status, the second status, and the resulting status are each one of the following: an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

9. The system for providing a floating point product of claim 8, wherein the overflow status represents one in a group of a plus overflow (+OV) status and a minus overflow (-OV) status.

10. The system for providing a floating point product of claim 8, wherein the overflow status is represented as a predetermined non-infinity numerical value.

11. The system for providing a floating point product of claim 8, wherein the underflow status represents one in a group of a plus underflow (+UN) status and a minus underflow (-UN) status.

12. The system for providing a floating point product of claim 8, wherein the underflow status is represented as a predetermined non-zero numerical value.

13. The system for providing a floating point product of claim 8, wherein the invalid status represents a not-a-number (NaN) status due to an invalid operation.

14. The system for providing a floating point product of claim 8, wherein the infinity status represents one in a group of a positive infinity status and a negative infinity status.

15. A method for providing a floating point product, comprising:

determining a first status of a first floating point operand and a second status of a second floating point operand based upon data within the first floating point operand and data within the second floating point operand respectively; and

asserting a resulting floating point operand containing the product of the first floating point operand and the second floating point operand and a resulting status embedded within the resulting floating point operand.

16. The method for providing a floating point product of claim 15, wherein the determining stage further comprises:

storing the first floating point operand in a first operand buffer;

storing the second floating point operand in a second operand buffer;

generating a first characteristic signal representative of the first status; and

generating a second characteristic signal representative of the second status.

17. The method for providing a floating point product of claim 16, wherein the first characteristic signal and the second characteristic signal are generated without regard to memory storage external to the first operand buffer and the second operand buffer.

18. The method for providing a floating point product of claim 17, wherein the memory storage external to the first operand buffer and the second operand buffer is a floating point status register.

19. The method for providing a floating point product of claim 15, wherein the asserting stage further comprises:

producing the product of the first floating point operand and the second floating point operand; and

asserting the resulting floating point operand having the resulting status embedded within the resulting floating point operand.

20. The method for providing a floating point product of claim 15, wherein the product of the first floating point operand and the second floating point operand is identical in all cases to the product that would be produced if the two operands were first swapped.

21. The method for providing a floating point product of claim 15, wherein the first status, the second status, and the resulting status are each one of the following: an

invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

22. The method for providing a floating point product of claim 21, wherein the overflow status represents one in a group of a plus overflow (+OV) status and a minus overflow (-OV) status.

23. The method for providing a floating point product of claim 22, wherein the overflow status is represented as a predetermined non-infinity numerical value.

24. The method for providing a floating point product of claim 21, wherein the underflow status represents one in a group of a plus underflow (+UN) status and a minus underflow (-UN) status.

25. The method for providing a floating point product of claim 24, wherein the underflow status is represented as a predetermined non-zero numerical value.

26. The method for providing a floating point product of claim 21, wherein the invalid status represents a not-a-number (NaN) status due to an invalid operation.

27. The method for providing a floating point product of claim 21, wherein the infinity status represents one in a group of a positive infinity status and a negative infinity status.

28. A computer-readable medium on which is stored a set of instructions for providing a floating point product, which when executed perform stages comprising:

determining a first status of a first floating point operand and a second status of a second floating point operand based upon data within the first floating point operand and data within the second floating point operand respectively; and

asserting a resulting floating point operand containing the product of the first floating point operand and the second floating point operand and a resulting status embedded within the resulting floating point operand.

29. The computer-readable medium of claim 28, wherein the determining stage further comprises:

storing the first floating point operand in a first operand buffer;

storing the second floating point operand in a second operand buffer;

generating a first characteristic signal representative of the first status; and

generating a second characteristic signal representative of the second status.

30. The computer-readable medium of claim 29, wherein the first characteristic signal and the second characteristic signal are generated without regard to memory storage external to the first operand buffer and the second operand buffer.

31. The computer-readable medium of claim 30, wherein the memory storage external to the first operand buffer and the second operand buffer is a floating point status register.

32. The computer-readable medium of claim 28, wherein the asserting stage further comprises:

producing the product of the first floating point operand and the second floating point operand; and

asserting the resulting floating point operand having the resulting status embedded within the resulting floating point operand.

33. The computer-readable medium of claim 28, wherein the product of the first floating point operand and the second floating point operand is identical in all cases to the product that would be produced if the two operands were first swapped.

34. The computer-readable medium of claim 28, wherein the first status, the second status, and the resulting status are each one of the following: an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

35. The computer-readable medium of claim 34, wherein the overflow status represents one in a group of a plus overflow (+OV) status and a minus overflow (-OV) status.

36. The computer-readable medium of claim 35, wherein the overflow status is represented as a predetermined non-infinity numerical value.

37. The computer-readable medium of claim 34, wherein the underflow status represents one in a group of a plus underflow (+UN) status and a minus underflow (-UN) status.

38. The computer-readable medium of claim 37, wherein the underflow status is represented as a predetermined non-zero numerical value.

39. The computer-readable medium of claim 34, wherein the invalid status represents a not-a-number (NaN) status due to an invalid operation.

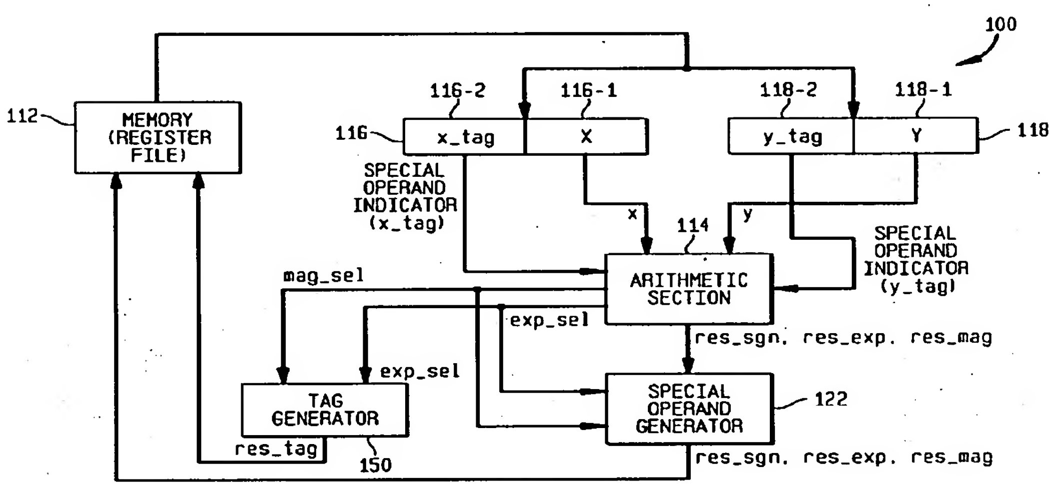
40. The computer-readable medium of claim 34, wherein the infinity status represents one in a group of a positive infinity status and a negative infinity status.

## X. Evidence Appendix to Appeal Brief Under Rule 41.37(c)(1)(ix)

Appellant relies on Fig. 4 of *Huang*, which is reproduced below.

**U.S. Patent** Nov. 30, 1999 Sheet 3 of 3 **5,995,991**

*FIG. 4*



## XI. Related Proceedings Appendix to Appeal Brief Under Rule 41.37(c)(1)(x)

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,747, in which an Appeal Brief was filed concurrently herewith.

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,595, in which an Appeal Brief was filed concurrently herewith.

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,584, in which an Appeal Brief was filed concurrently herewith.

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,587, in which an Appeal Brief was filed concurrently herewith.

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,647, in which an Appeal Brief was filed concurrently herewith.

The Appeal of In re Application of Guy STEELE, Jr., U.S. Patent Application No. 10/035,579, in which an Appeal Brief was filed on concurrently herewith.